

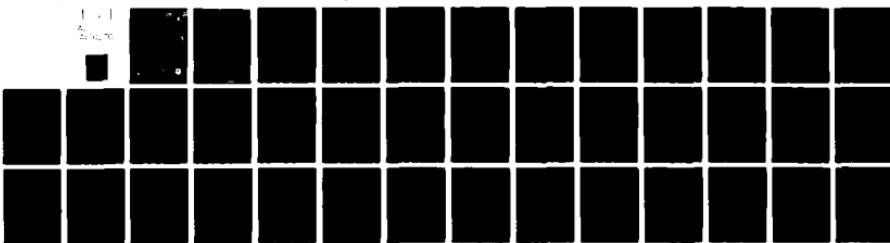
AD-A100 270

AIR FORCE GEOPHYSICS LAB HANSCOM AFB MA
MICROPROCESSOR CONTROLLED PULSE AMPLITUDE MODULATION DECOMMUTAT--ETC(U)
OCT 80 K R WALKER, J R GRIFFIN

F/G 9/6

UNCLASSIFIED AFGL-TR-80-U335

NL



END
DATE
FILED
7 81
DTIC

LEVEL II

(12)



**AFGL-TR-80-0335
INSTRUMENTATION PAPERS, NO. 286**

Microprocessor Controlled Pulse Amplitude Modulation Decommutator

**KENNETH R. WALKER
JACK R. GRIFFIN**

DTIC
ELECTED
S JUN 16 1981 D
E

28 October 1980

Approved for public release; distribution unlimited.

DTIC FILE COPY

**AEROSPACE INSTRUMENTATION DIVISION
AIR FORCE GEOPHYSICS LABORATORY
HANSOM AFB, MASSACHUSETTS 01731**

PROJECT 7659

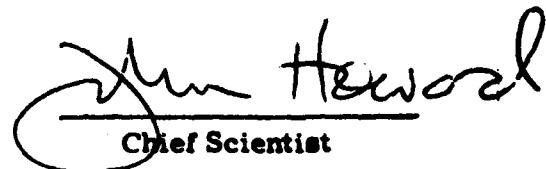
AIR FORCE SYSTEMS COMMAND, USAF



This report has been reviewed by the ESD Information Office (OI) and is releasable to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



John Heward
Chief Scientist

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

(9) J. L. Walker, Jr.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFGL-TR-80-0335.	2. GOVT ACCESSION NO. AD-A100270	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MICROPROCESSOR CONTROLLED PULSE AMPLITUDE MODULATION DECOMMUTATOR	5. TYPE OF REPORT & PERIOD COVERED Scientific. Interim.	
7. AUTHOR(s) Kenneth R. Walker Jack R. Griffin	6. PERFORMING ORG. REPORT NUMBER IP No. 295	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Geophysics Laboratory (LCR) Hanscom AFB Massachusetts 01731	8. CONTRACT(S) OR GRANT NUMBER(S) AFGL-TR-XA-7-35 AFGL-TR-80-0335	
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratory (LCR) Hanscom AFB Massachusetts 01731	10. PROGRAM ELEMENT, PROJECT, TASK AREA AND WORK UNIT NUMBERS 62101F 76590104	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE 28 October 1980	
	13. NUMBER OF PAGES 41	
	15. SECURITY CLASS (of this report) Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Microprocessor Pulse amplitude modulation Software phase-locked loop IRIG commutators PAM decommutation		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the design of a microprocessor-controlled pulse amplitude modulation decommutator. The microprocessor hardware design and software routines are covered in detail. The decommutation technique provides flexibility in output formatting such as: a. visual display of decommutated data reduced to engineering units with labels, flags, and time; b. a printer output for hard copy of visual display; c. analog signal outputs for strip chart recording.		

DD FORM 1 JAN 73 EDITION OF 1 NOV 85 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Will lend/or	
Dist	Initial
A	

Contents

1. INTRODUCTION	5
2. OBJECTIVE	6
3. BACKGROUND	6
4. DESIGN	7
4.1 Desired Performance	7
4.2 Hardware	8
4.2.1 Development System	8
4.2.2 Interface	9
4.3 Software	17
4.3.1 Background Job	18
4.3.1.1 BASIC Interpreter	18
4.3.1.2 BASIC Program	18
4.3.1.3 PAM Decom Program	20
4.3.2 Foreground Job	23
5. DISCUSSION OF RESULTS	32
6. CONCLUSION	33
APPENDIX A: BASIC PAM Program	35
APPENDIX B: Sample Output	39
LIST OF ABBREVIATIONS	41

Illustrations

1. System Block Diagram	9
2. Tape Cassette Interface	10
3. Video Terminal Interface	10
4. Analog to Digital Converter Interface	10
5. BCD Time Code Generator Interface	12
6. Thumbwheel Switch Interface	13
7. Analog Output Latch/DAC	15
8. ACIA Printer Driver	15
9. Address Decoding	15
10. System Memory Map	17
11. Program Modules	18
12. Background Job Flowchart	19
13. Sample PAM Wavetrain	21
14. Output Scaling for D/A Ports	22
15. Foreground Job Flowchart	24
16. Sync Detection Flowchart	25
17. Phase Lock Flowchart	28
18. List and Definition of Variables	30
19. Flag Bit Assignment	31
20. Phase Lock Samples	32

Microprocessor Controlled Pulse Amplitude Modulation Decommutator

I. INTRODUCTION

Telemetry data frequently consists of housekeeping monitors, such as temperatures and battery voltages, which convey the internal status of an airborne research payload. This information is often telemetered back to the ground in a PAM (Pulse Amplitude Modulated) format on one subcarrier oscillator data channel, rather than on a number of channels.

Present decommutation systems in use are hardwired logic boxes with limited flexibility and poor performance with noisy or certain types of data. Their output is a scaled analog voltage suitable only for driving a strip chart recorder or voltmeter.

A microprocessor-based system can provide a much more reliable unit as well as a more flexible device. Formats may be changed by simply using different software. The outputs may be scaled, labeled, and printed out to display the actual units of the parameter being measured.

The KIM-1 microprocessor evaluation board by MOS Technology is used as the building block to design the prototype system. Input, output, memory, and display devices are added as required.

A description of the prototype system hardware and software will be presented, as well as ideas for incorporation into the final model.

(Received for publication 16 October 1980)

2. OBJECTIVE

The object of this report is to present and describe the application of a microprocessor development board to the task of decommutation of a PAM data commutator. The techniques presented will provide the design background for the development of an intelligent decom which will perform the basic decom function, scale to engineering units, and display on a line printer the segment number, raw voltage value, label, and actual engineering units of the parameter being measured.

3. BACKGROUND

Pulse Amplitude Modulation (PAM) is a technique of encoding a number of analog signals on a single analog line by sequentially sampling each individual signal. The amplitude of the information channel pulse then becomes the analog-variable parameter. This process of encoding is known as commutation, or time division multiplex. The encoded signal is then normally applied to a Sub-Carrier Oscillator (SCO) and then mixed with other SCO's in a standard FM Multiplex Telemetry Link.

This report is concerned specifically with a 30-channel commutator which samples each channel 2.5 times per second. Other standard formats are also in use.¹ There are two standard output formats for the output signal: Return to Zero (RZ) and Non-Return to Zero (NRZ). The output format affects the synchronization pattern and the duty cycle of the channel signal. This report deals only with the NRZ format.

The synchronization signal of a 2.5×30 NRZ commutator is specified as follows by IRIG-106:¹

<u>Channel No.</u>	<u>Voltage</u>
26	Zero amplitude
27	Full scale amplitude
28	Full scale amplitude
29	Full scale amplitude
30	50 percent full scale or less

Channels 1 through 25 are defined as data channels.

For purposes of this report, zero amplitude is defined as 0 V dc, and full scale amplitude is defined as 5 V dc. This voltage is compatible with standard commutators and SCO's. All inputs to the commutator must be scaled, usually in a linear relationship, between these two values.

1. IRIG Document 106-80 Standards for Telemetry Systems

Decommutation is the process of reconstructing the PAM wavetrain. This process includes synchronization, channel selection, and a sample and hold process for the selected channel(s). To improve this process, software will be added to control the entire process, to scale to original engineering units, label, and to print out the data on a video terminal and line printer.

Several problems have existed in the past with the hardware oriented decommutators:

1. Difficulty in reconstructing the timing clock. This is especially difficult on an NRZ signal where the clock frequency is not inherent in the wavetrain. The common technique is to use a triggered oscillator or phase locked loop, but a series of data channels with equal voltage levels will often cause the system to lose lock.
2. False sync patterns. It is quite possible that a series of data channels may look like the sync pattern during some phase of the data collection. Judicious assignment of data channels can help alleviate this problem, but is not always successful.
3. Data comprehension. When used in a decision-making process during long test sequences, the problem of sorting out, scaling, and labeling the data in a meaningful format becomes significant.
4. Capacity. Most decons will display only a limited number of channels at one time. Display of all significant channels would be desirable.

4. DESIGN

4.1 Desired Performance

The following features are considered essential to the satisfactory performance of this PAM decommutator:

1. Format: 30-segment NRZ commutator.
2. Channel Rate: 75 Hz, that is, sample rate = 2.5/s.
3. Input Level: ± 5 V peak with automatic scaling of any level less than ± 5 V.
4. A phase-lock loop synchronization for control of sampling times.
5. Sixteen analog outputs to display data on strip chart recorders.
6. Front panel control of the channel numbers to be output to the 16 analog outputs.
7. Front panel control and display of a selected channel.
8. ASCII output to a CRT or hard-copy printer device.
9. External input for time of year data to be used as a label on hard-copy printout.

10. ASCII keyboard input to facilitate programming of output formatting (scaling, labeling, etc.).

11. Operation from a 117 V ac nominal power source.

4.2 Hardware

The hardware described in this section consists of a general purpose microprocessor development system and specific hardware for the decom application. The actual decom instrument will consist of only those parts required and constructed specifically for the application.

4.2.1 DEVELOPMENT SYSTEM

The MOS Technology KIM (Keyboard Interface Module) microprocessor evaluation board is used as a building block for the development system. This basic board contains a 6502 8-bit microprocessor, two 6530 RAM/ROM timer/interface chips, 1-K byte RAM, a keyboard, 6-digit decimal display, serial ASCII transmit/receive port, and tape recorder input and output ports. One of the 6530 chips contains a monitor program in ROM which interfaces and controls the various functions and allows the user to display memory addresses and change their contents. Additional description of this board and a listing of the monitor program is contained in the KIM-1 User Manual.²

KIM is plugged into a 'KIMSI' mother board which provides buffering of all microprocessor address, data, and control lines. It converts the KIM control signals to S-100 signals, and provides a standard S-100 bus format with expansion/interface 100 pin connectors.³

Standard 8-K byte memory boards are plugged in to provide RAM as necessary. Other boards, such as EPROM, input, and outboard boards are also plugged into the mother board to make a complete system.

Communication with the development system is through a standard 20-mA loop serial ASCII keyboard and display such as a Teletype® or video terminal. This interface is part of the KIM evaluation board. The KIM onboard display and keyboard are not used when the external ASCII terminal is used.

The KIM board also provides a tape recorder interface which is used as permanent mass program storage during the development process.

The power supply consists of two high-frequency switching supplies that provide 5 V dc at 10 A and ±15 V dc at 1.5 A for distribution to all subsystems via the mother board.

2. KIM-1 User Manual, Mos Technology Inc., Publication No. 6500-15B.

3. KIMSI Assembly/Operating Manual, Forethought Products.

4.2.2 INTERFACE

The various input and output circuits are described in this section. Figure 1 is a block diagram showing the configuration of these various interface circuits. The address, data, and control busses are considered as part of the development system, and appear on the mother board. Detailed description and drawings of the interface circuits are shown up to the bus or KIM application connector, as appropriate, with references to the pin numbers and function.

The cassette recorder, used for program storage, and the video terminal, used for input output operations, connect directly to the KIM board and are described in the KIM-1 manual.² Interface cabling is shown in Figures 2 and 3.

The analog to digital converter accepts the amplitude-modulated wavetrain from the discriminator and converts it, under processor control, to a digital count each time it is commanded to convert by the decom program. The converter used is an Analog Devices model ADC10Z 10-bit converter, from which only the 8 most significant bits are used. It accepts a bipolar signal between -5 and +5 V and outputs a proportional count between 0 and 255 ($2^8 - 1$). The converter is connected directly to parallel interface lines on the KIM board as shown in Figure 4.

The total dynamic errors inherent in an FM multiplex system (that is, those that cannot be reduced or eliminated by calibration) are generally considered to be

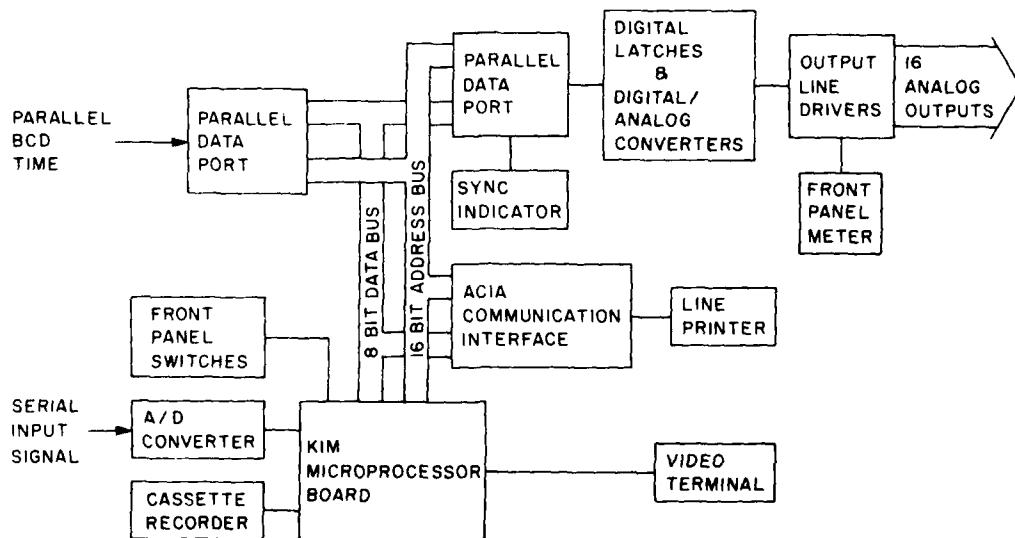


Figure 1. System Block Diagram

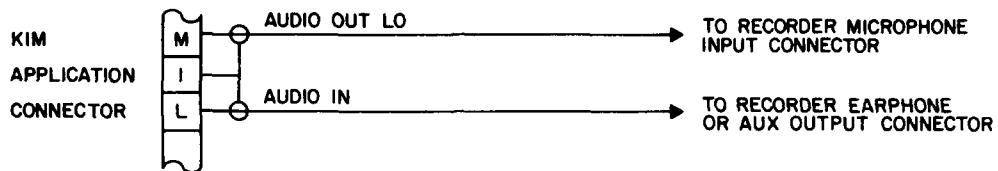


Figure 2. Tape Cassette Interface

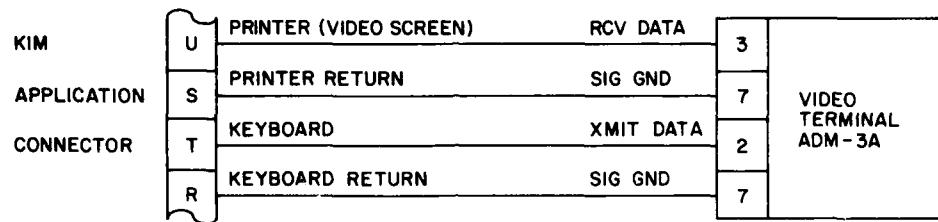


Figure 3. Video Terminal Interface

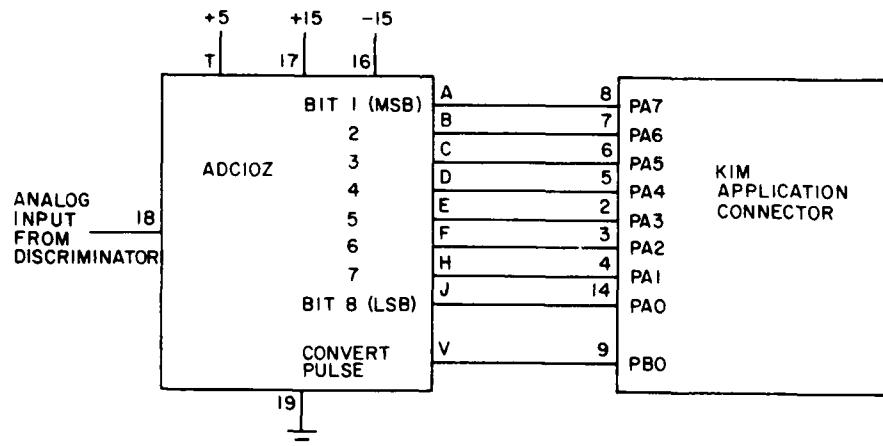


Figure 4. Analog to Digital Converter Interface

approximately 2 percent, or 100 mV for the 0 to 5 V input signal.⁴ The resolution of an 8-bit system digitizing a 5 V signal is approximately 20 mV, well within the overall system accuracy. Even when the discriminator voltage swing is reduced by one half, resulting in the use of only one half of the available digital counts, the resolution of the encoded signal becomes 40 mV, still well within overall system accuracy. This is important since the software allows the discriminator output to be set at any voltage swing between ± 5 and ± 2.5 V with no effect on system calibration and negligible effect on overall system resolution. It should be noted that an input signal that exceeds ± 5 V is out of range of the converter and is not acceptable.

The time code generator used in this application is a DATUM model 9310 with computer interface. This provides time of day information for labeling the data output, and for correlation purposes later.

The time code generator is interfaced to the microprocessor through a 6520 peripheral interface adapter port. See Figure 5. In addition to the 16 time of day lines, 3 port lines are provided to control the computer interface on the generator. Time is presented as two parallel 16-bit words to the port under software control of the three control lines. The processor reads the port 8 bits at a time and stores the data in memory for further processing.

The time code generator is valuable for non-real time data printout since the IRIG B signal from a data tape may be translated, read by the processor, and printed out with the data.

The front panel thumbwheel switches are used to select one data segment to be displayed on the front panel digital meter. The selected segment will also appear on analog output No. 1. See Figure 5 for interface wiring. A low on the PB1 line selects the MSD switch and a positive true BCD code appears on PB2-PB5. A high on PB1 selects the LSD in the same way.

The analog output ports, Figure 6, consist of a 6520 peripheral interface port, a latch clock line decoder/selector, data latches for each output port, 8-bit digital to analog converters, and analog output drivers. Two output ports are shown in Figure 7. Additional latches are connected in parallel with those shown, and the appropriate latch clock line is connected to the 4515 clock line selector. Up to 16 analog output lines may be connected in this manner. The front panel digital meter is connected to port number 1 and displays whichever segment has been selected for that port by the front panel thumbwheel switches.

The output drivers are connected to rear panel connectors and may be used to drive meters, ink pen recorders, or light beam recorders. Nominal output is 0 to 5 V or 0 to 20 mA.

4. EMR Telemeter No. 13 Aug 1976.

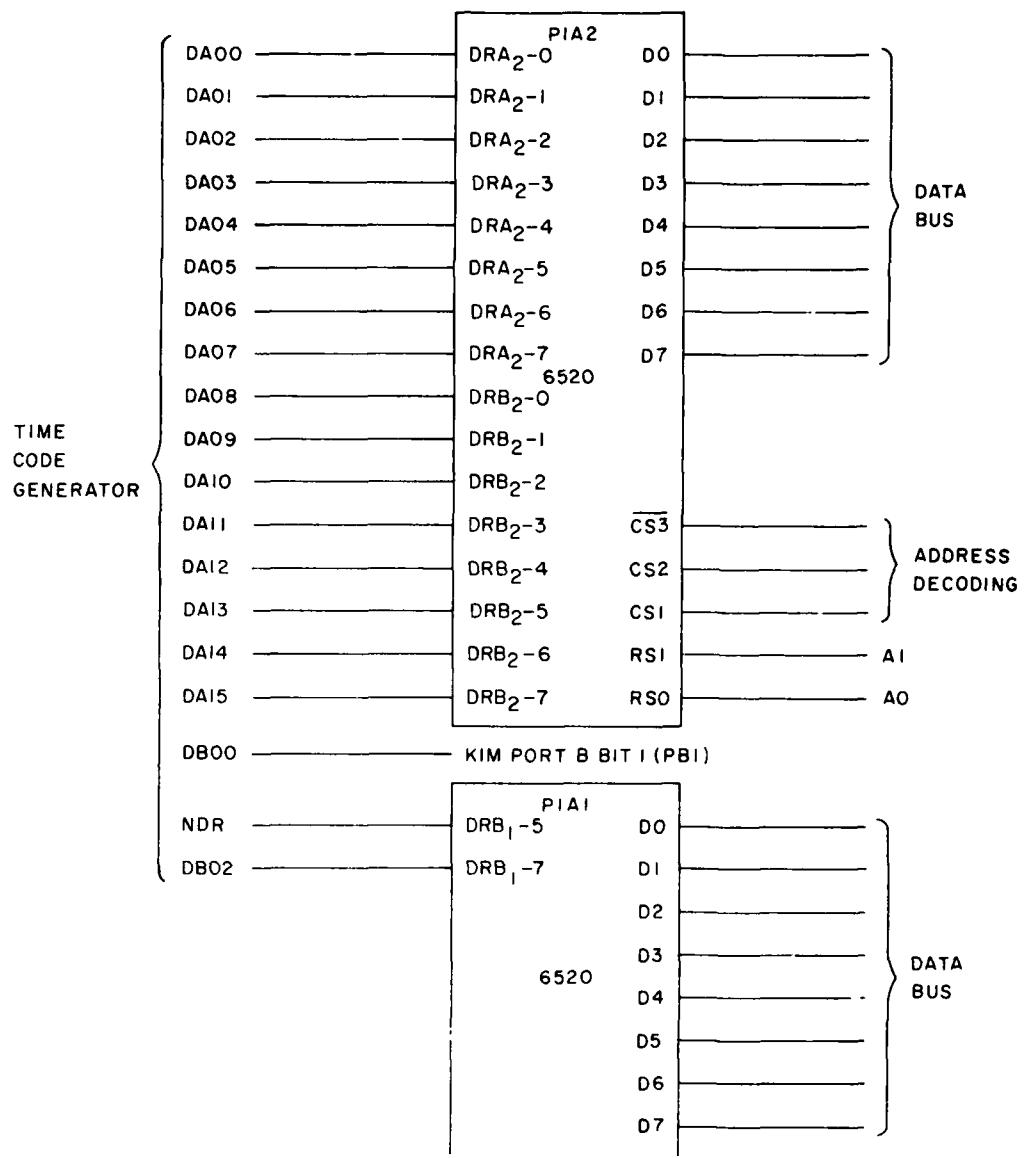


Figure 5. BCD Time Code Generator Interface

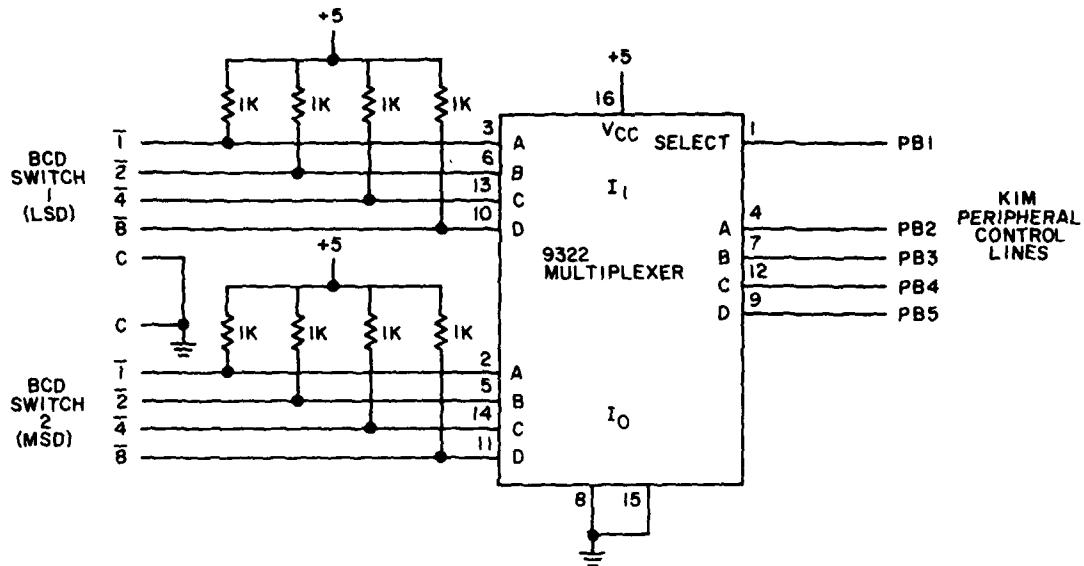


Figure 6. Thumbwheel Switch Interface

The sync light is used to indicate when the decom has synchronized with the sync pattern, and is controlled by the logic level on data register B, bit 6 of the 6520 PIA 1 chip.

The Asynchronous Communication Interface Adapter (ACIA) in Figure 8 is a bus-oriented device, that is, it connects directly to the processor bus. It is used to provide the data formatting and control to interface serial asynchronous data communications information to the microprocessor. The configuration of Figure 8 is for serial ASCII output to a 20-mA loop device, such as a printer. The ACIA accepts ASCII-encoded parallel data from the bus and places it in a buffer. It then adds the required start and stop bits and outputs a serial bit stream at a baud rate determined by the bit rate generator. Bits are set in the internal status register to prevent entering new data before the current character has been transmitted. Through this device the printer provides a permanent record of all commutator segments in both raw voltage levels and scaled engineering units. Time tags and appropriate headings are added to each page of data.

The video terminal, Figure 3, is interfaced directly to the KIM board in a 20-mA loop configuration. The terminal is used to communicate with the KIM monitor upon power up, and provides an input device for initializing the system. When the decom program is run, the screen becomes an output device for the commutator data.

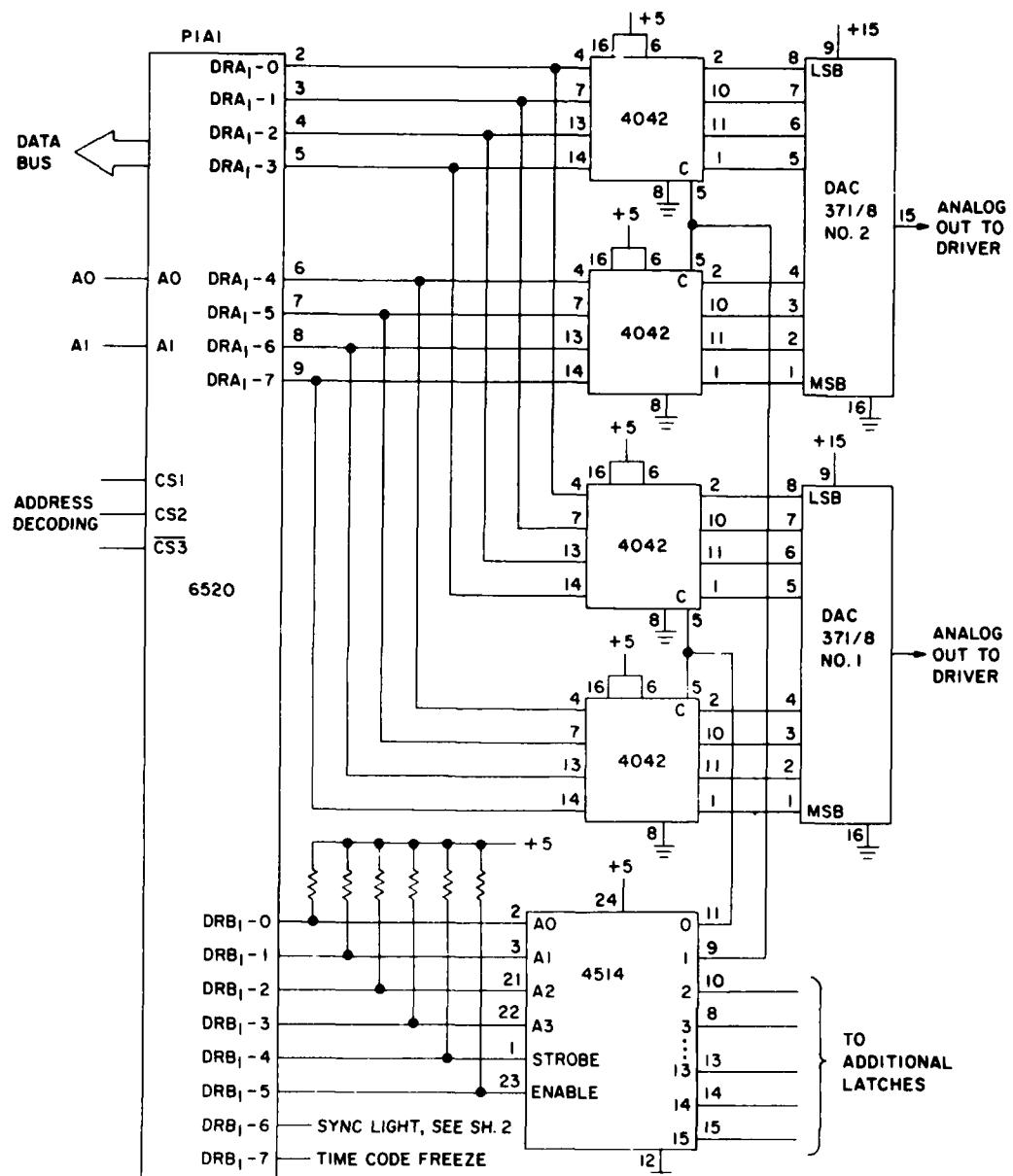
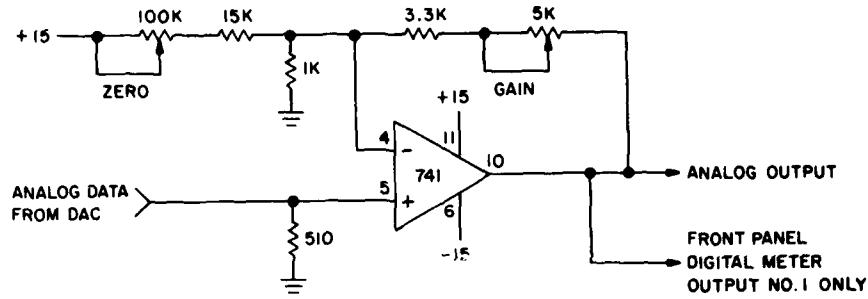


Figure 7. Analog Output Latch/DAC (Sheet 1)



TYPICAL ANALOG OUTPUT DRIVER

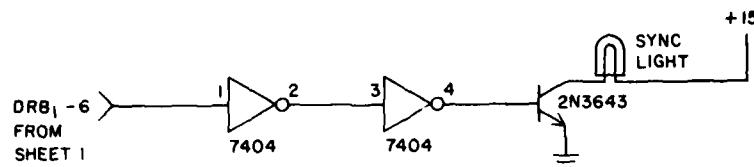


Figure 7. Analog Output Latch/DAC
(Sheet 2)

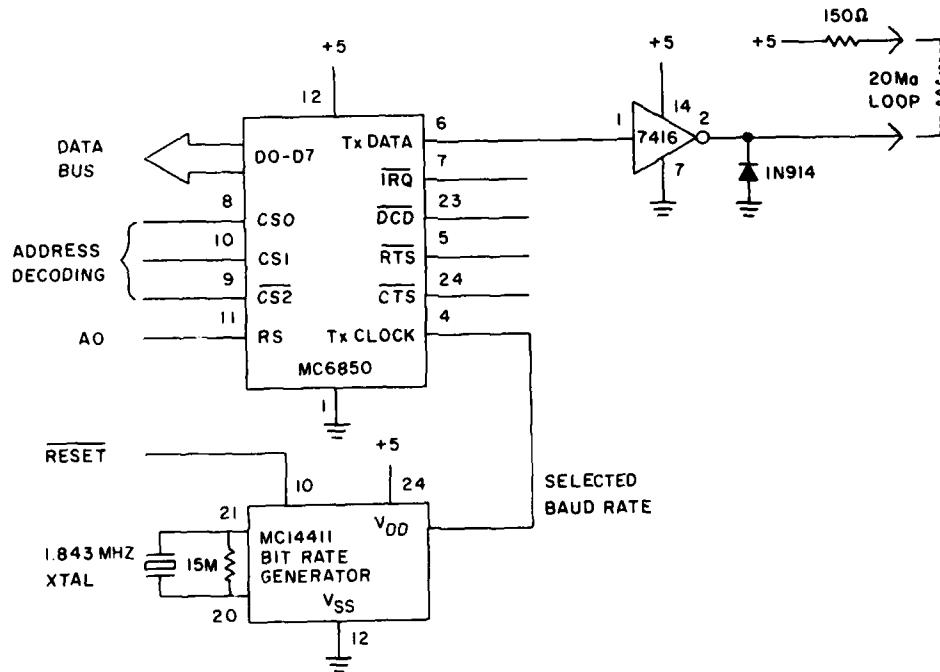


Figure 8. ACIA Printer Driver

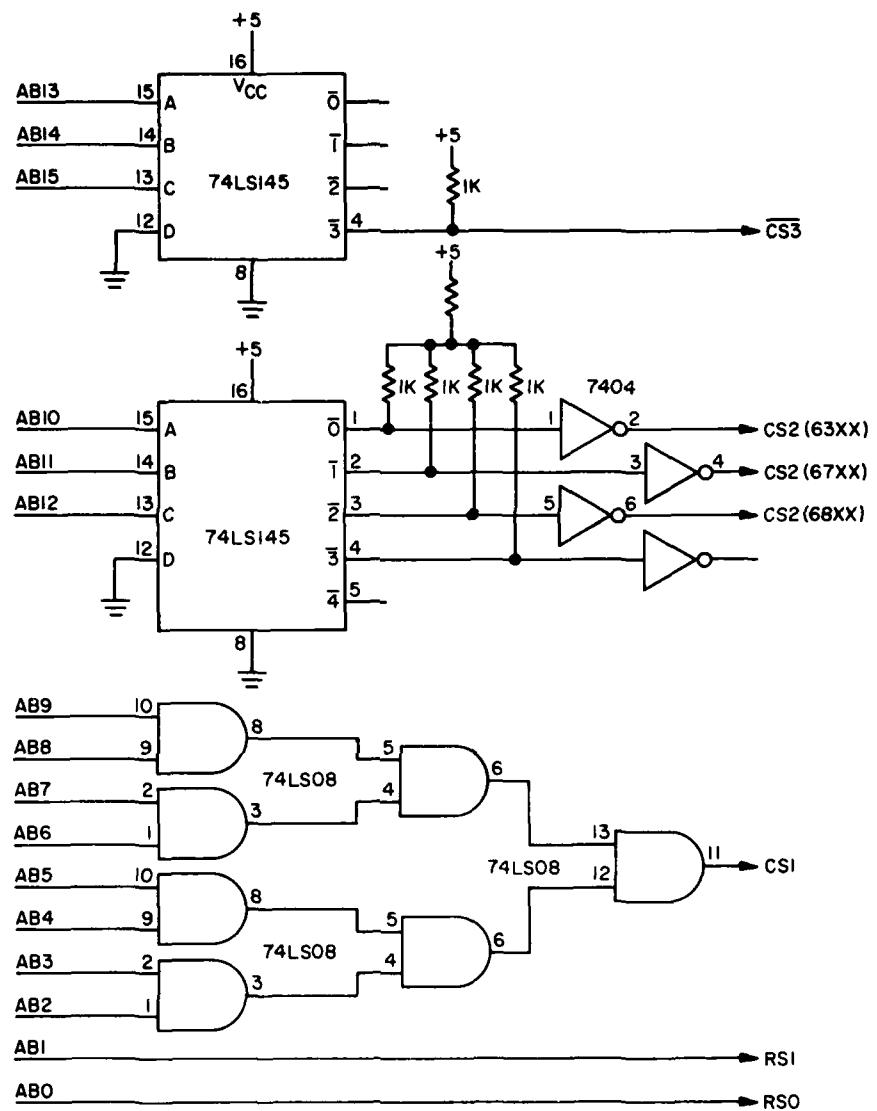


Figure 9. Address Decoding

The KIM board provides for storage and retrieval of programs on magnetic tape format, such as a common cassette recorder. Most of the program may be stored in EPROM and will be available upon power up. The BASIC program which controls, scales, labels, and formats the data output is subject to frequent change, and therefore is stored on tape and loaded into the RAM each time the system is initialized. The combination of EPROM and RAM storage results in a very flexible system. Data format changes may be made using a BASIC program stored on tape while those routines not subject to change are programmed permanently in the system.

All devices in a 6502 processor system are assigned an address. This process of addressing a device is known as memory-mapped I/O. Since the microprocessor chip has 16 address lines available to address any one of the 64-K addresses, additional decoding is required to uniquely define a device. Since many systems do not require the entire 64-K addresses, blocks of address may be assigned to a device in order to simplify decoding. See Figure 9 for the address decoding circuit and Figure 10 for a memory map of the entire system.

<u>HEX Address</u>	<u>Assignment</u>
E000-FFFF	Not Used
C000-DFFF	EPROM (PAM Decom)
8000-BFFF	Not Used
6000-7FFF	Input/Output Devices
2000-5FFF	EPROM/RAM (BASIC)
1800-1FFF	KIM ROM
1700-17FF	KIM I/O & Timers
0400-16FF	Not Used
0200-03FF	RAM
0100-01FF	STACK
0000-00FF	Zero Page

Figure 10. System Memory Map

4.3 Software

The software for this system may be divided into two very distinct sections: the timing-dependent "foreground job," and the "background job," which is running whenever the foreground job is not. These two sections are further broken down into modular parts in the following discussion. See Figure 11.

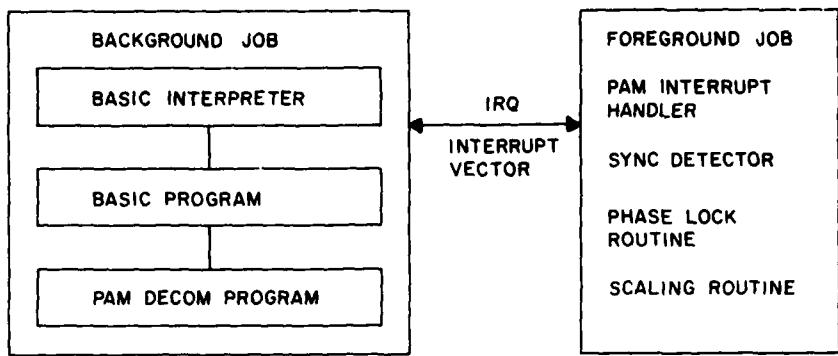


Figure 11. Program Modules

4.3.1 BACKGROUND JOB

The background job is the program that controls the entire system operation. It initializes the system upon power on and directs the program flow through the various modules. Whenever the system is reset, control returns to the background job. See Figure 12.

4.3.1.1 BASIC Interpreter

The BASIC interpreter is a program which was purchased from Microsoft Inc. for the KIM-1 system to support and execute programs written in BASIC. It was incorporated into this system to simplify the programming requirements for scaling, labeling, and formatting the data output to the printer. Various modifications have been made to it to support the printer, to allow it to operate in the foreground/background mode, and to allow it to be placed in nonvolatile memory (EPROM).

4.3.1.2 BASIC program

The BASIC interpreter interprets and executes the BASIC program on a line by line basis. See Appendix A for a listing of the BASIC program used on Project BAMM. The program calls a series of routines which actually execute the decom program, read the time of day clock, read in the stored data segment values, and drive the printer. Data segments may be selectively displayed as required, labels and scaling factors may be easily modified, and limit flags are displayed on the printout where appropriate. Hard copy printout times may be selected at any interval, or at specific times as required. The entire system may be applied to a different project or one with multiple commutators simply by changing the BASIC program. Appendix B contains sample data printout as a result of the program in Appendix A. The use of a high-level language such as BASIC in conjunction with

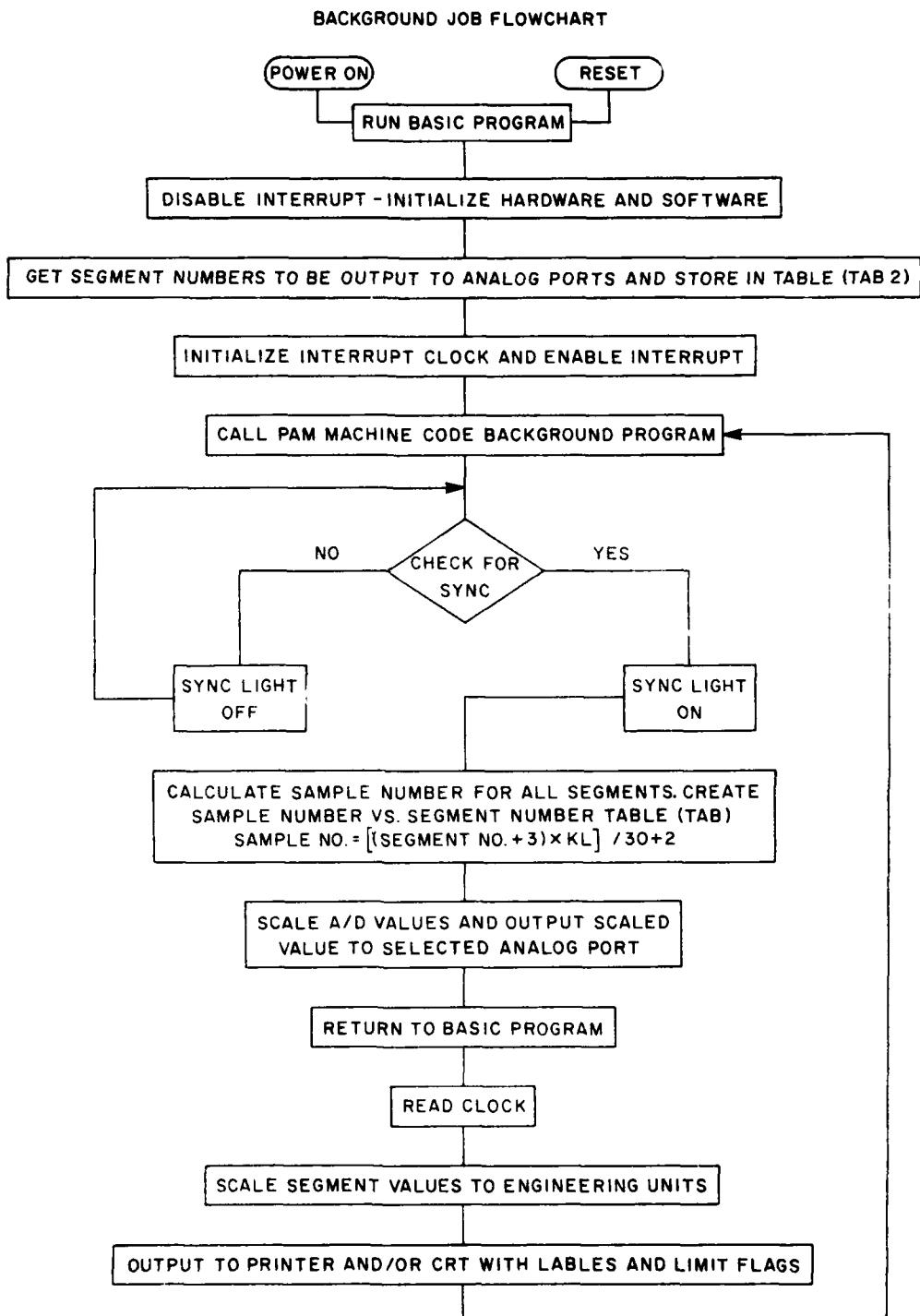


Figure 12. Background Job Flowchart

machine language routines results in a very effective compromise of flexibility (BASIC) and operating speed (machine code) required in a real-time data acquisition environment.

4.3.1.3 PAM Decom Program

The PAM decom program is actually a machine-code level subroutine that is called from the BASIC program. This program and the interrupt-driven foreground program are the heart of this report.

The decom program module does the non-time-dependent segments of the decommutation task, including the following:

1. Initialize hardware and software.
2. Request entry of up to 16 segment numbers for output to analog output ports.
3. Check Phase Lock status.
4. Control Phase Lock indicator.
5. Calculate sample counter number corresponding to desired segments and store them in a table.
6. Create a table of scaled output values for output to hardware output ports.
7. Output scaled values to analog output ports.
8. Control interrupt enable/disable flag.

None of the above segments are time dependent (that is they may be done whenever there is time available after returning from the foreground job).

The PAM decom program module waits in a loop until the Phase Lock status flag has been set by the foreground job. Once lock has been established, the frame length counter is used to determine a matrix of sample numbers corresponding to the requested segment number. Refer to Figure 13 during the following discussion.

For a nominal 2.5×30 NRZ commutator, the number of samples per frame (KL), at a 3-ms sample rate, would be:

$$2.5 \text{ frames/s} = 400 \text{ ms/frame}$$

$$\text{Frame length} = \frac{400 \text{ ms}}{\text{frame}} \times \frac{1 \text{ sample}}{3 \text{ ms}} = 133 \text{ samples/frame} .$$

This also gives a segment sample rate of:

$$\frac{400 \text{ ms}}{\text{frame}} \times \frac{1 \text{ sample}}{3 \text{ ms}} \times \frac{1}{30 \text{ segments}} = \frac{4.444 \text{ samples}}{\text{segment}} .$$

The sample number corresponding to a segment number is then found as follows:

$$\text{Sample No.} = \frac{[(\text{segment no.} + 3) \times \text{KL}]}{30} + 2 .$$

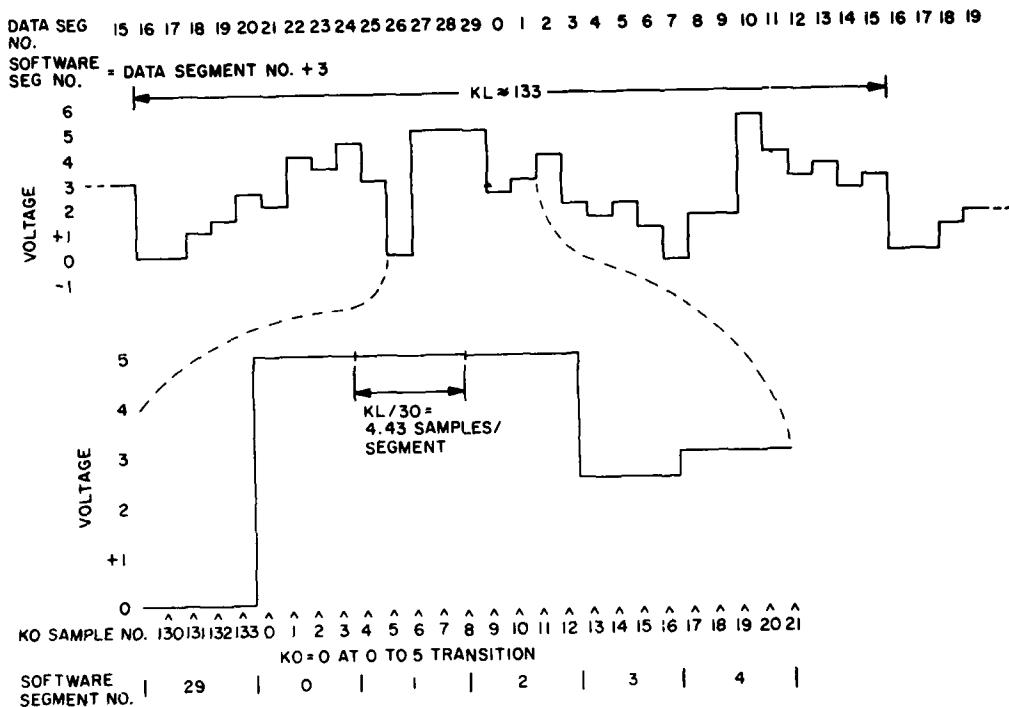


Figure 13. Sample PAM Wavetrain

For example, given:

Frame length $KL = 133$
Segment No. 1

$$\text{KO Sample No.} = \frac{(1 + 3) \times 133}{30} + 2 = 19.73 \approx 19$$

For the purposes of the calculations, the fractional part of a result is dropped since sample numbers are only meaningful in integer values. Sample No. 19 would then correspond with the value of segment no. 1. All segments are sampled toward the end of the segment period to allow for slow rise time in the telemetry system and to allow any ringing to be damped out. Figure 13 shows the relation between frame length, software segment number, data segment number, and sample number, with an expanded view of the sync pattern.

After the sample number matrix has been created, the sampled segment values are scaled for output such that digital output counts are equivalent to the following values:

$$1 \text{ count} = 27.3 \text{ mV}$$

<u>Count</u>	<u>Voltage</u>
00	-1.00
36	0.00
127	2.50
219	5.00
255	+6.00

Figure 14 also shows this relationship in graphic form. This range exceeds the normal commutator voltage by ± 20 percent, but occasional out of range values have been experienced, and the output device should reflect this fact.

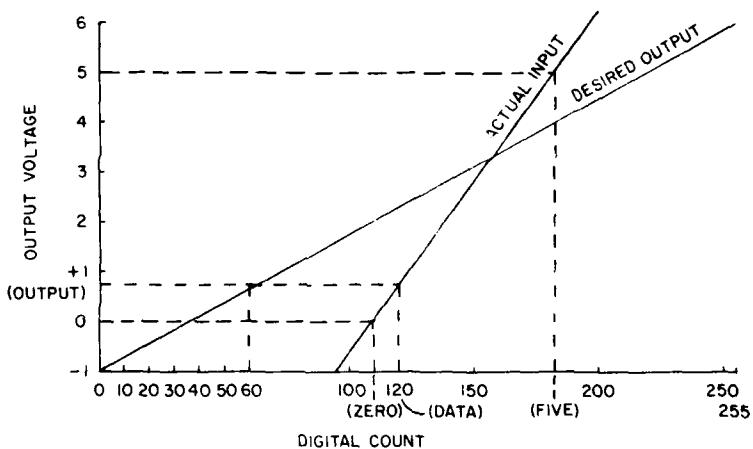


Figure 14. Output Scaling for D/A Ports

The scaling routine is a proportional relationship based on the input A/D count determined by the foreground job at the 0 and 5-V points.

The following relationship is used:

$$\text{D to A} = \frac{(219 - 36) \times (\text{Data} - \text{Zero})}{(\text{Five} - \text{Zero})} + 36$$

where

Zero = A/D counts at 0-V input

Five = A/D counts at 5-V input

Data = A/D counts at desired segment number.

As an example, and referring to Figure 14, given:

Zero = 110

Five = 182

Data = 120

$$D \text{ to } A = \frac{(219 - 36) \times (120 - 110)}{(182 - 110)} + 36 = 61 .$$

A digital count of 61 sent to the appropriate digital to analog output device will result in an output voltage of 0.69 V. All requested segment numbers are scaled and sent to their respective output port in turn.

4.3.2 FOREGROUND JOB

The PAM foreground job, commonly referred to as the interrupt service routine, executes the time-dependent parts of the decom task. Once the system has been initialized and interrupts are allowed to occur (that is, the interrupt line is enabled), a hardware timer counts down from a predetermined value and causes an interrupt pulse to be generated every 3 ms. The interrupt pulse causes the processor to interrupt the task being performed and transfer control to the foreground program. All background code is reentrant, that is, it may be interrupted and then resumed after servicing the interrupt without losing any intermediate data or affecting the results. Figures 15, 16, and 17 show the sequence of events that occur in response to the interrupt. Figure 18 is a list of variables that should be referred to during the following discussion.

Before the system can find sync, it must take 255 samples, or almost two complete frames, in order to determine the highest and lowest input values with which it must work. Four sequential samples are averaged and used to update HI and LO. These two values may be, but are not necessarily, the values corresponding to the 0 and 5-V segments. Once HI and LO are found, a flag is set and all further input values are expanded such that HI equals a count of 255 and LO equals a count of 0. It is this feature that allows the input to the system to be set over a wide voltage range with no effect on the output data.

The program now proceeds to check subsequent samples for a valid sync pattern, as shown in Figure 16. Once a required voltage level or transition is detected, the associated flag bit is set and the program flow returns to the background program.

FOREGROUND JOB FLOWCHART

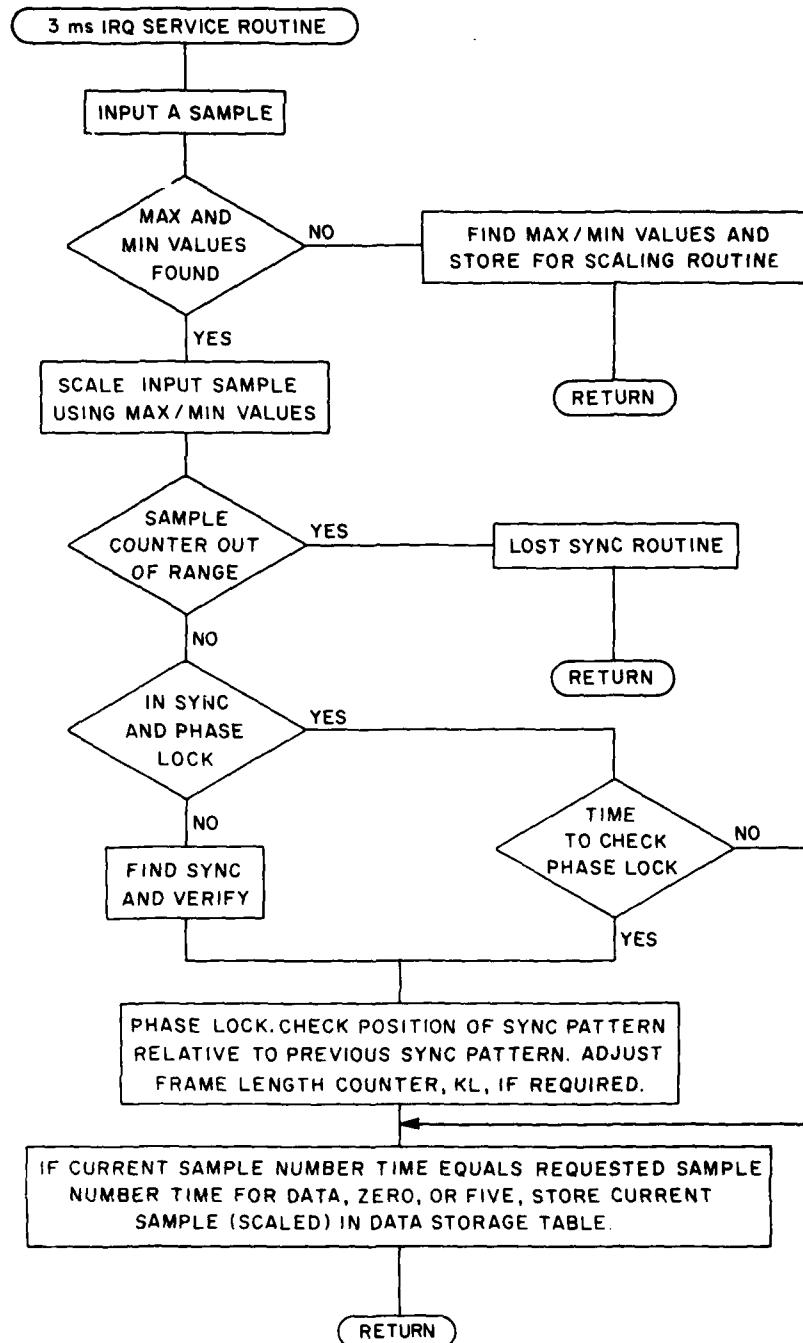


Figure 15. Foreground Job Flowchart

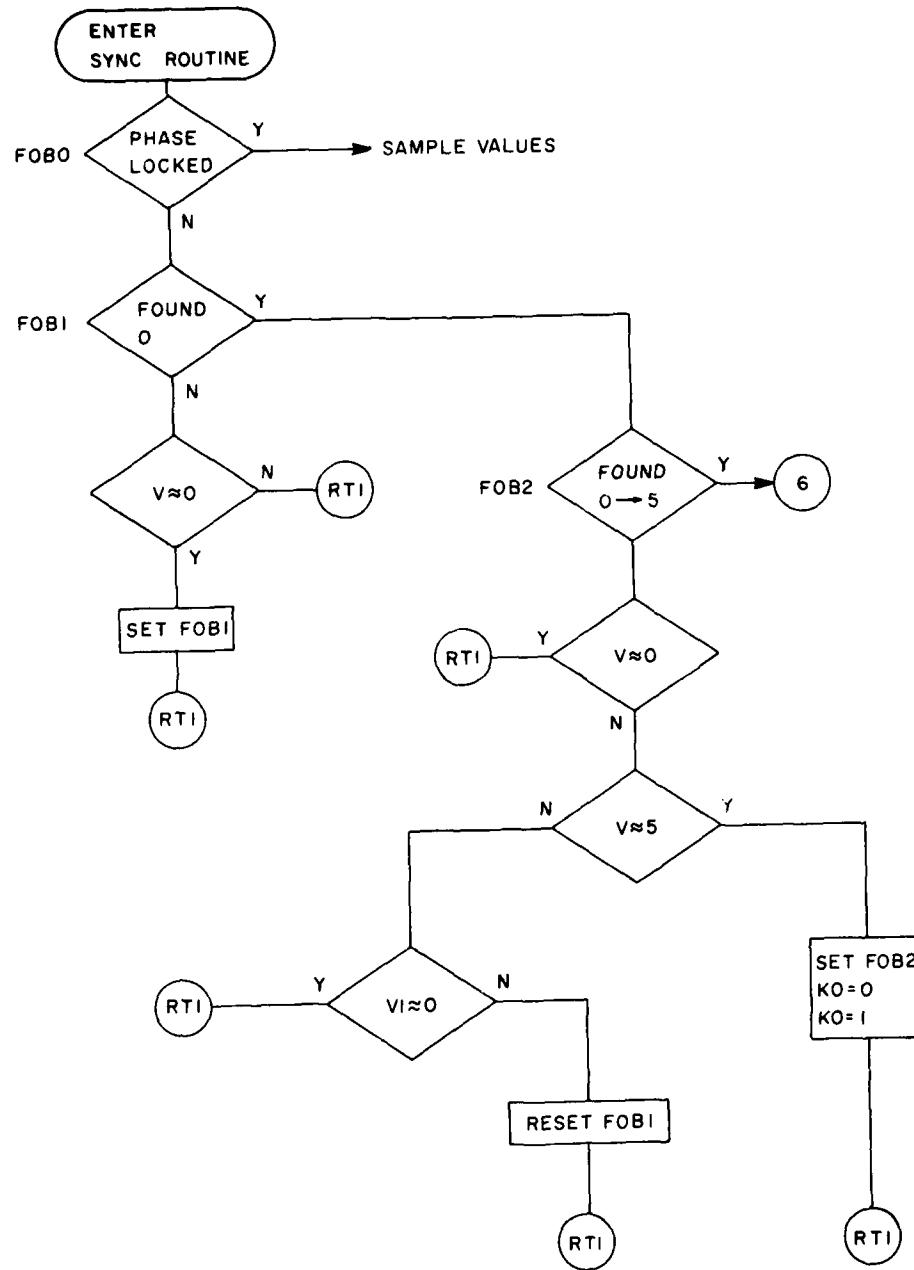


Figure 16. Sync Detection Flowchart (Sheet 1)

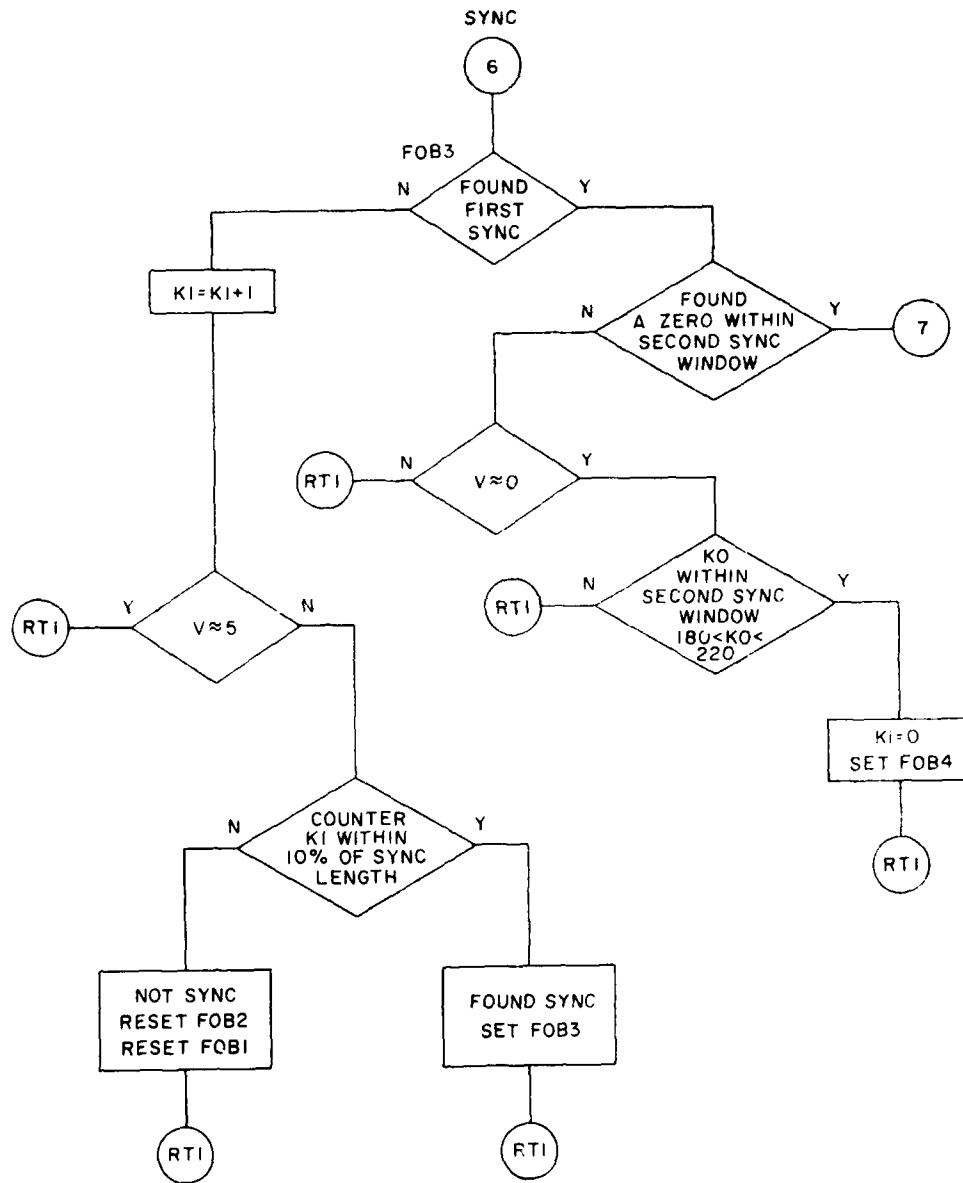


Figure 16. Sync Detection Flowchart (Sheet 2)

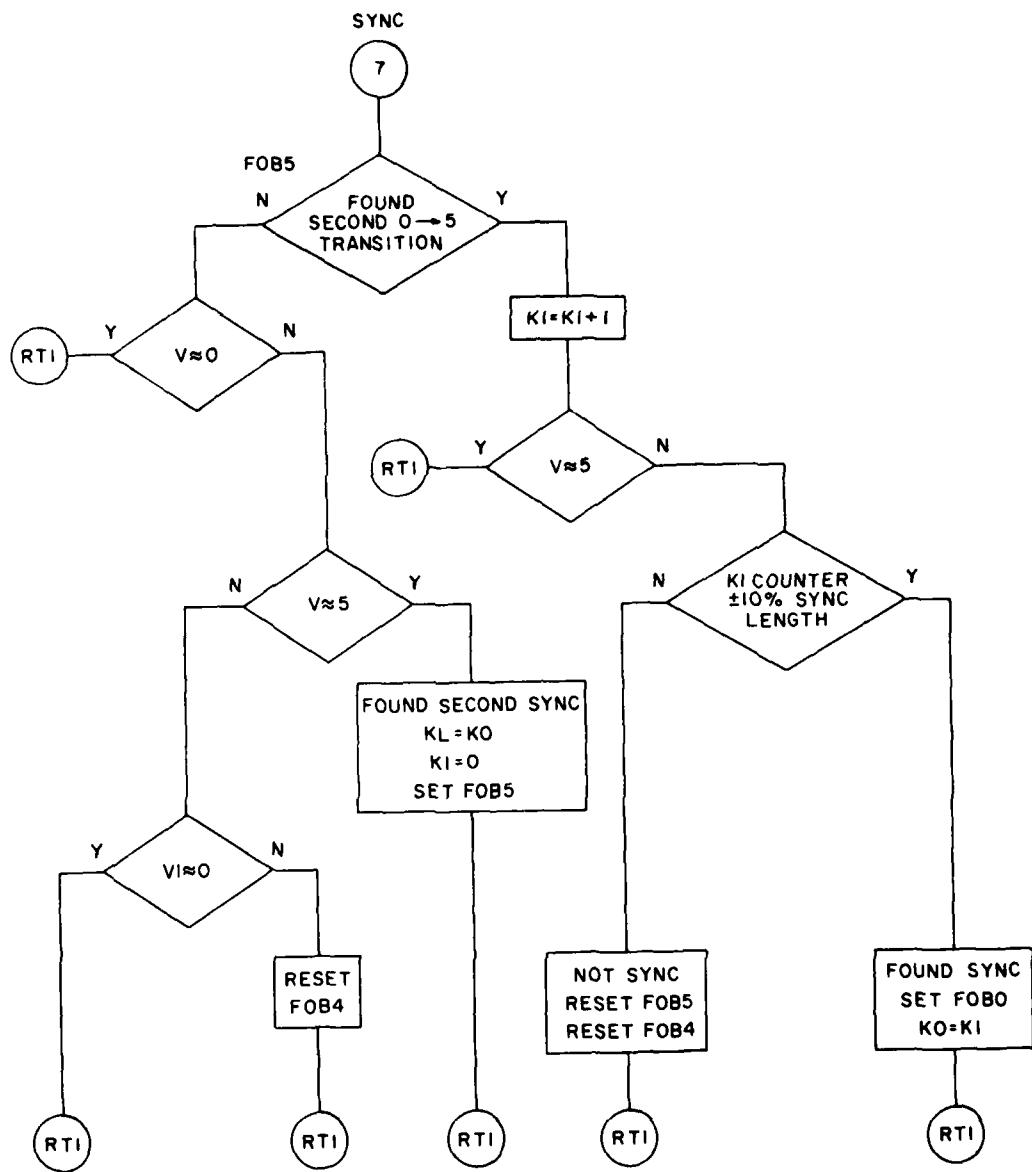


Figure 16. Sync Detection Flowchart (Sheet 3)

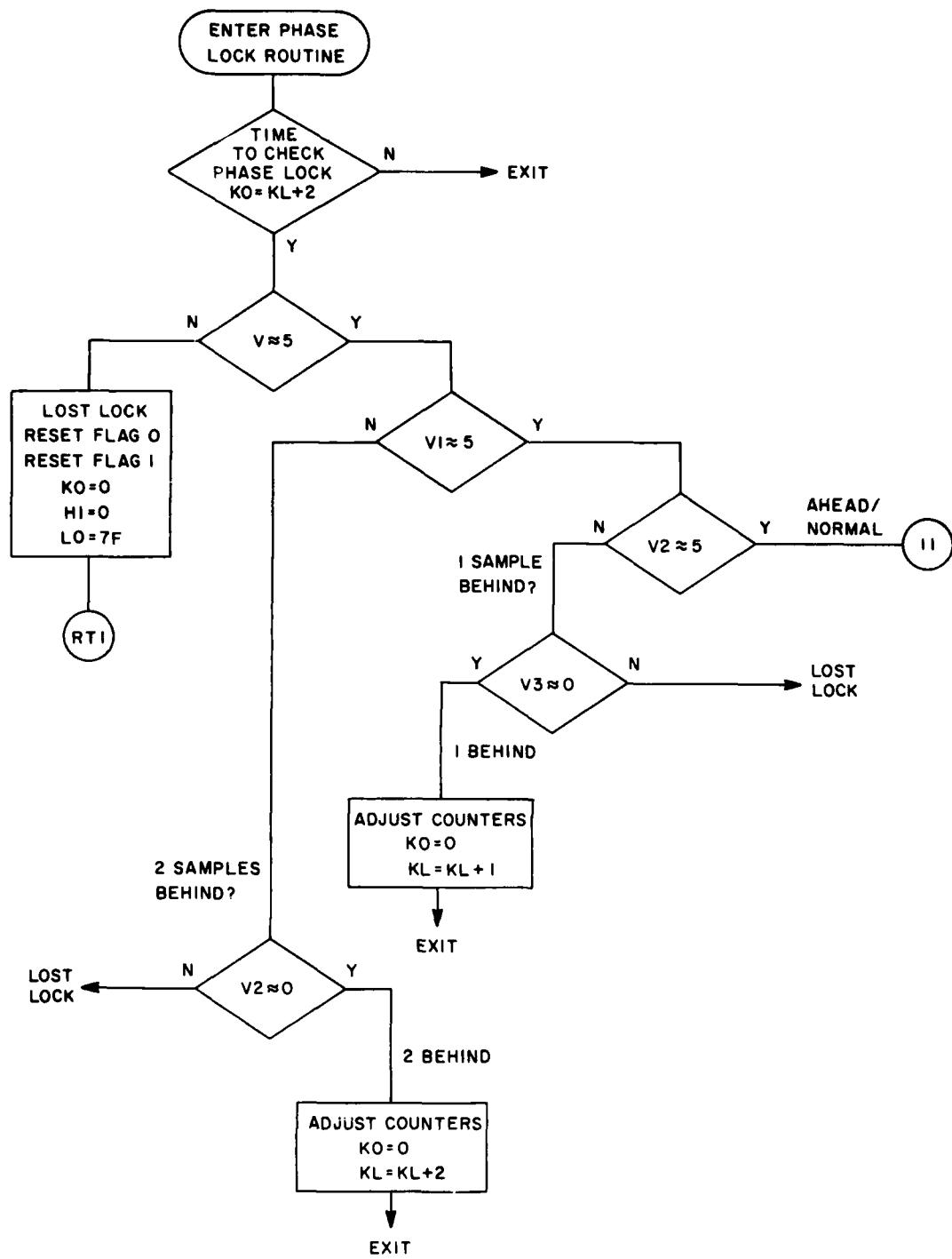


Figure 17. Phase Lock Flowchart (Sheet 1)

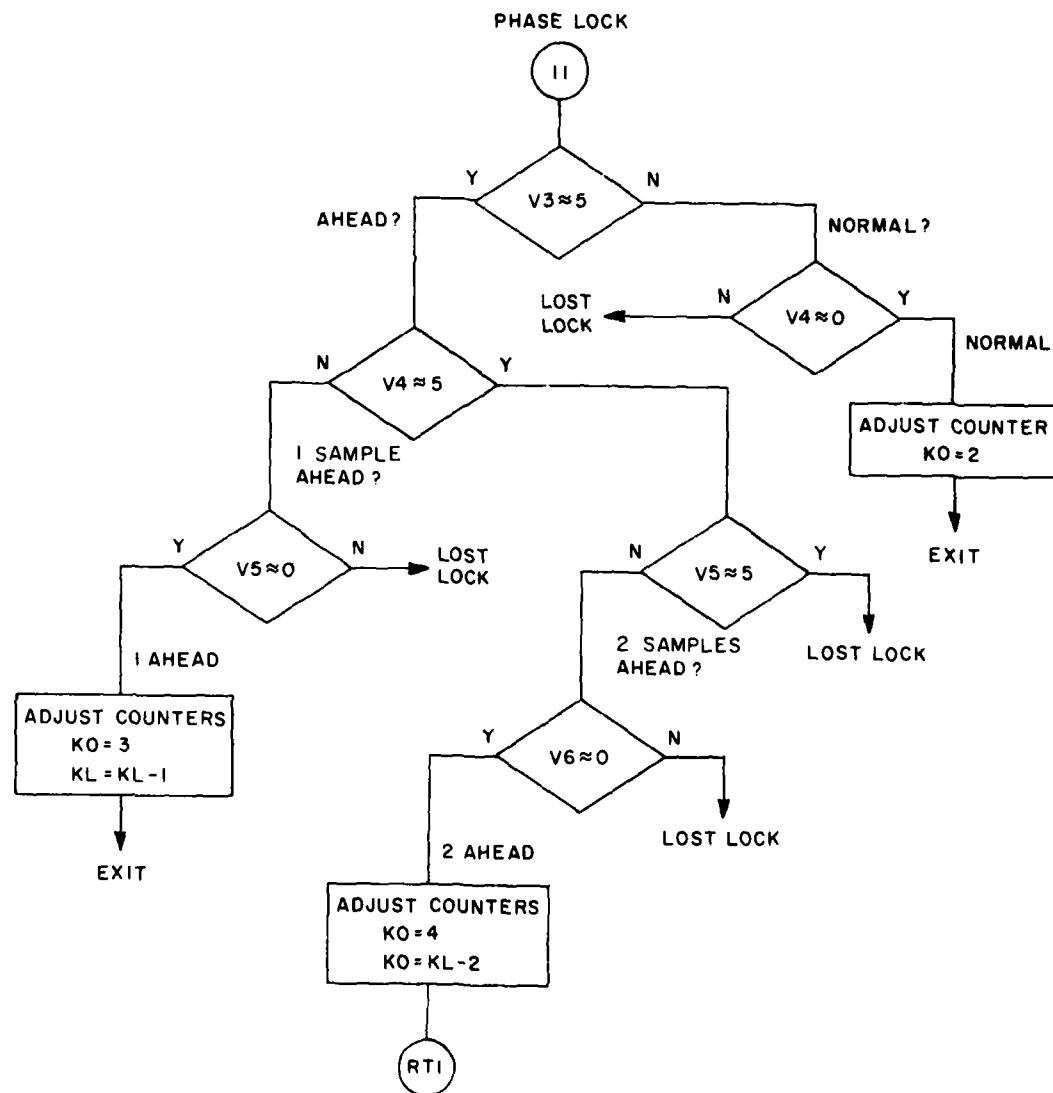


Figure 17. Phase Lock Flowchart (Sheet 2)

<u>Label</u>	
F0	Flagword 0
F1	Flagword 1
V	Current A/D input voltage (V(N))
V1	Previous input voltage (V(N-1))
V2	V(N-2)
V3	V(N-3)
V4	V(N-4)
V5	V(N-5)
V6	V(N-6)
K0	Interrupt counter
K1	Counter
KL	Counts between sync (frame length)
ZERO	A/D at 0-V segment
FIVE	A/D at 5-V segment
ZERO0	Last zero
FIVE0	Last five
KZERO	Clock at zero
KFIVE	Clock at five
LO	Smallest A/D
HI	Largest A/D
RAW	Raw A/D value
RAWLO	Raw LO A/D value
RAWHI	Raw HI A/D value
KLO	Old value of K1.

Tables

TAB2	Desired segment numbers
TAB	Clock count to sample TAB2
DATA	A/D count corresponding to TAB
DTOA	D/A VALUE (corrected/scaled) for output

Figure 18. List and Definition of Variables

Upon reentry at the next interrupt, the program checks its progress by testing the flags, and proceeds directly to the next step in the sync detection process. If at any time a particular test fails, it is either checked again at the next interrupt time, or the flags are reset and the process begins again. Figure 19 lists the flag bits and their significance.

Flag Word 0

Bit

0	Phase Locked
1	Found a Zero
2	Found a Zero to Five Transition
3	Found First Sync Pattern
4	Found a Zero, possible start of second sync
5	Found Second Zero to Five Transition
6	Not Used
7	Not Used

Flag Word 1

Bit

0	Taken 255 samples and found LO and HI
1	Not Used
2	Found LO and HI and initialized K0 Counter
3	Not Used
4	Not Used
5	Not Used
6	Found First Zero
7	Found First Five

Figure 19. Flag Bit Assignment

Once a second sync word is found within a 10 percent window of one frame length from the first sync word, the system indicates sync lock and proceeds to process data. This window allows up to a 10 percent variation in the sampling rate of the commutator before it will drop sync. Present-day solid state commutators, unlike the earlier mechanical variety, do not exhibit these large speed changes, but this allows the system to process data from any commutator running within ± 10 percent of the 2.5×30 rate.

Once each frame the relative position of the sync word 0 to 5-V transition is compared with that of the last frame. This is the phase lock program module shown in Figure 17. The program allows and will correct for a change of ± 2 interrupt pulses, or 6 ms, before it drops back to the sync detection routine.

Figure 20 shows the acceptable range of phase lock and the corrections applied in each case. If the routine determines that the previously measured frame length was too long, it will subtract 1 or 2 from the frame length counter KL. If the frame length was too short, 1 or 2 will be added to the counter as appropriate. At

the same time, the sample or interrupt counter K0 is preset to the correct sample number. A change of more than ± 2 is considered unacceptable, and the system drops out of sync and begins the process again. This change in KL is taken into account on each pass through the background PAM routine when the sample number vs. segment number table (TAB) is created.

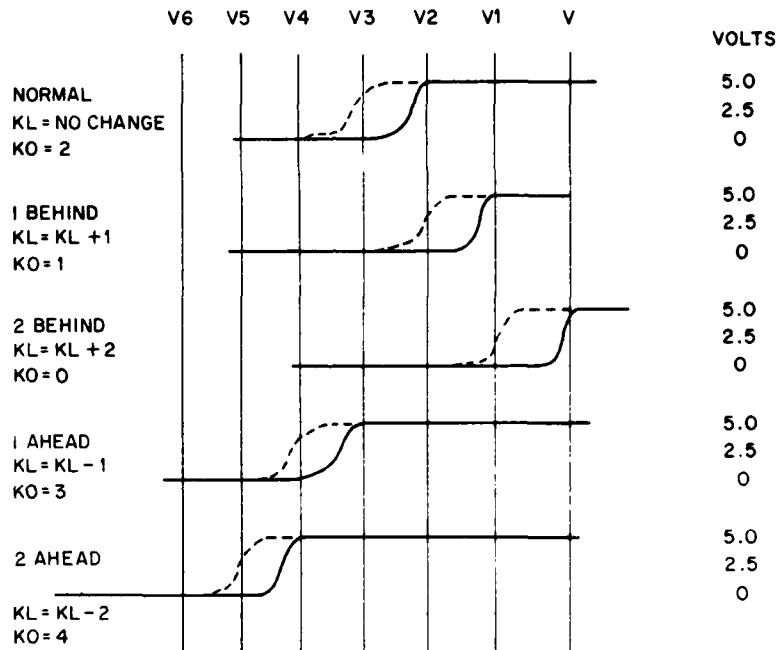


Figure 20. Phase Lock Samples

5. DISCUSSION OF RESULTS

A working prototype system has been designed, built and field-tested during a series of high-altitude balloon launches for project BAMM. The system has also been used for data reduction for post-flight analysis. This experience has pointed out where various changes could be made to the system. As with many systems involving software, there always seems to be another small change available to make the system a little better. The major software change to be made involves a query from the video terminal or a front panel switch which will bypass the BASIC program and output to the video terminal and printer. This will allow the system

to be used as a standalone decommutator to drive a paper chart recorder. Up to 16 segment numbers would be entered from a front panel numeric keypad.

Other changes involve hardware simplification. Since this project was started, advances in A/D and D/A conversion hardware have been made which would simplify the interface by allowing these devices to be connected directly to the processor bus.

The final model should not use a KIM-board, but would contain a processor board specifically designed for the system. This would eliminate parts of KIM which are not used, such as the keyboard and display.

Address decoding could be simplified to decode only those blocks of address that would be used in the final system. Complete decoding would not be necessary.

Finally, the system could be constructed in a cabinet sized to hold a common plug in card size to facilitate changes and servicing. The resident software in EPROM could be on one card which would be easily interchanged when software modifications are to be made.

A final model is presently under design which will include these changes. This model will be assembled and tested as time permits.

6. CONCLUSION

Construction of this development and prototype system has demonstrated the feasibility of applying a microprocessor to the field of telemetry data handling. It also shows the power and versatility available from a combination of hardware and software. This project has been an intriguing learning experience into the field of microprocessors. The knowledge and experience gained during this project will be applied to further the state of the art of telemetry system support in the future.

Appendix A

BASIC PAM PROGRAM

```

10 GOSUB 2000
20 DIM D(29):REM A-D COUNTS
25 DIM L$(29): REM LABELS
26 DIM AS(1): REM DAY / DATE STRING
28 DIM V(29):REM SCALED 0-5 VOLTS
30 POKE 8256,203:POKE 8257,193
35 X=USR(0)
40 PRINT
41 PRINT
42 PRINT
45 LET AS= " WEDNESDAY 8 OCTOBER 1980"
50 PRINT TAB(5); "BAM-1 PAYLOAD STATUS",AS
60 PRINT
62 PRINT
65 LET Z=PEEK(944)
66 LET F=PEEK(945)
70 FOR I= 0 TO 29
80 LET A=PEEK(828+I)
85 LET D(I)=A
90 GOSUB 1000
95 LET V(I)=A
100 READ L$(I)
110 NEXT
115 IF V(1)=0 THEN IF V(2)=0 THEN PRINT TAB(15); "SIMULATED DATA"
116 IF V(1)>0 THEN PRINT
118 PRINT
210 LET A=V(1)*2.85+19
212 GOSUB 1010
218 PRINT " 01 "V(1);TAB(15);L$(1)="A" VOLTS "
220 LET A=V(2)*2.54+20.51
222 GOSUB 1010
228 PRINT " 02 "V(2);TAB(15);L$(2)="A" VOLTS "
240 LET A=V(4)
242 GOSUB 1110
244 GOSUB 1010
246 GOSUB 2400
248 PRINT " 04 "V(4);TAB(15);L$(4)="A" C ",T" F "
250 LET A=V(5)
252 GOSUB 1110
254 GOSUB 1010
256 GOSUB 2400
258 PRINT " 05 "V(5);TAB(15);L$(5)="A" C ",T" F "
260 LET A=V(6)
262 GOSUB 1110
264 GOSUB 1010
266 GOSUB 2400
268 PRINT " 06 "V(6);TAB(15);L$(6)="A" C ",T" F "
270 LET A=V(7)
272 GOSUB 1110
274 GOSUB 1010
276 GOSUB 2400
278 PRINT " 07 "V(7);TAB(15);L$(7)="A" C ",T" F "
300 IF V(17)<.2 THEN GOTO 2300
360 LET A=V(16)
362 GOSUB 1110
364 GOSUB 1010
366 GOSUB 2400
368 PRINT " 16 "V(16);TAB(15);L$(16)="A" C ",T" F "
370 LET A=V(17)*4.88-3.73
372 GOSUB 1010
378 PRINT " 17 "V(17);TAB(15);L$(17)="A" PSI "

```

```

380 IF V(19)<.2 THEN GOTO 2200
390 LET A=V(19)*(-21.614)+258.07
392 GOSUB 1010
398 PRINT " 19 "V(19);TAB(15);LS(19)="A"K "
400 LET A=V(20)*(-18.66)+30.1
402 GOSUB 1010
408 PRINT " 20 "V(20);TAB(15);LS(20)="A"C "
410 LET A=V(24)*(-9.66)+121.4
412 GOSUB 1010
418 PRINT " 24 "V(24);TAB(15);LS(24)="A"K "
420 LET A=V(25)*(-16)+49
422 GOSUB 1010
428 PRINT " 25 "V(25);TAB(15);LS(25)="A"C "
430 PRINT
432 PRINT "      SYNC"; V(26); V(27); V(28); V(29); V(0); "
434 RESTORE
436 GOTO 30
1000 LET A=5*(A-Z)/(F-Z)
1010 LET A=INT(A*100+.5)/100
1020 RETURN
1100 REM SUBROUTINE TO SCALE LCR TEMP MON.
1110 IF A<1 THEN LET A=105.34-(45.5*A):RETURN
1120 IF A<1.8 THEN LET A=86.95-(26.95*A):RETURN
1130 IF A<3.8 THEN LET A=75.62-(19.75*A):RETURN
1140 IF A>4.5 THEN LET A=109.42-(28.57*A):RETURN
1150 LET A=254.18-(60.6*A):RETURN
2000 REM SUBROUTINE TO CLEAR SCREEN
2010 POKE 8256,235:POKE 8257,193
2020 X=USR(0)
2030 RETURN
2200 IF V(20)<.2 THEN GOTO 2210
2205 GOTO 390
2210 IF V(24)<.2 THEN GOTO 2220
2215 GOTO 390
2220 IF V(25)<.2 THEN GOTO 2230
2225 GOTO 390
2230 PRINT " 19 "V(19);TAB(15);"RADIOMETER OFF"
2240 PRINT " 20 "V(20);"
2250 PRINT " 24 "V(24);"
2260 PRINT " 25 "V(25);"
2270 GOTO 522
2300 PRINT " 16 "V(16);TAB(15);"TV CAMERA OFF"
2310 PRINT " 17 "V(17);"
2320 GOTO 380
2400 REM SUB TO CONVERT C TO F
2410 LET T=(9*A)/5+32
2420 LET T=INT(T*10+.5)/10
2430 RETURN
3000 DATA SYNC
3010 DATA TM BATT #1,TM BATT #2,SPARE
3020 DATA LINK 1 TEMP,LINK 2 TEMP,LINK 3 TEMP,LINK 4 TEMP
3030 DATA SPARE,SPARE,SPARE,SPARE
3040 DATA SPARE,SPARE,SPARE,SPARE
3050 DATA TV CAMERA TEMP,TV CAMERA PRESSURE
3060 DATA SPARE,RAD DET TEMP,RAD BAFFLE TEMP
3070 DATA SPARE,SPARE,SPARE
3080 DATA RAD FINGER TEMP,RAD MOTOR TEMP
3090 DATA SYNC,SYNC,SYNC,SYNC
4000 END
OK

```

Appendix B

SAMPLE OUTPUT

PRECEDING PAGE BLANK-NOT FILLED

BAMM PAYLOAD STATUS
WEDNESDAY 3 OCTOBER 1979

276 : 13 : 50 : 11 CDT

01	3.17	TM BATT #1= 28.03 VOLTS	
02	2.91	TM BATT #2= 27.9 VOLTS	
03	2.62	CMND TEMP= 23.1 C	73.6 F
04	2.04	LINK 1 TEMP= 33.97 C	93.1 F
05	1.87	LINK 2 TEMP= 37.16 C	98.9 F
06	1.96	LINK 3 TEMP= 35.47 C	95.8 F
07	1.98	LINK 4 TEMP= 35.1 C	95.2 F
08	2.4	TM BATT #1 TEMP= 27.22 C	81 F
13	2.47	INTER BATT #1= 28.38 VOLTS	
14	1.85	MONITOR TEMP= 37.53 C	99.6 F
15	2.32	INTER BATT #2= 32.58 VOLTS	
16	.02	TV CAMERA OFF	
17	0		
18	3.4	RAD BATT= 27.64 VOLTS	
**	19	- .36 RAD DET TEMP= 265.85 K	
	20	- .36 RAD HAFFLE TEMP= 309.82 K	
**	24	- .36 RAD FINGER TEMP= 124.88 K	
	25	1.09 RAD FILTER TEMP= 304.56 K	

SYNC 0 5.06 5.02 5 2.51

List of Abbreviations

ACIA	- Asynchronous Communications Interface Adapter
A/D	- Analog to Digital Conversion
ASCII	- American Standard Code for Information Interchange
BAMM	- Balloon Altitude Mosaic Measurement
BCD	- Binary Coded Decimal
CRT	- Cathode Ray Tube
D/A	- Digital to Analog Conversion
EPROM	- Eraseable-Programmable Read Only Memory
I/O	- Input/Output
K	- 1024; KBYTE = 1024 memory bytes = 1024×8 bits
KIM	- Keyboard Interface Module
LSD	- Least Significant Digit
MSD	- Most Significant Digit
NRZ	- Non-Return to Zero
PAM	- Pulse Amplitude Modulation
PIA	- Peripheral Interface Adapter
ROM	- Read Only Memory
RZ	- Return to Zero
SCO	- Sub-Carrier Oscillator